

# WebDAV and VOSpace

Walter Landry  
IRSA/IPAC/Caltech



# WebDAV

- Makes the web into a readable and writable medium
- Operates over HTTP and HTTPS
- Adds a few HTTP verbs
  - PROPFIND (find properties)
  - PROPPATCH (edit properties)
  - MKCOL (mkdir)
  - COPY
  - MOVE
  - ...

# WebDAV Standards

- IETF standard
  - RFC 2518 published in 1999
  - RFC 4918 published in 2007
- Other standards build upon WebDAV: CalDAV, GroupDAV, DeltaV
- IANA defines dav: as a Permanent URI Scheme

# Widely Implemented

- Modules for every major web server: Apache, nginx, IIS
- Standalone servers written in Python, Erlang
- Clients built into desktop operating systems
  - Drag and drop from your file browser
- Free clients available for everything down to phones.
- Client libraries available in many languages: C, Java, Python, Perl
- Command line clients: curl, cadaver

# Widely Implemented

- CERN implemented a WebDAV front end to their grid storage systems, partially replacing GridFTP.

<https://dfs.cern.ch/>

# VOSpace vs WebDAV

- VOSpace
  - add or delete data objects
  - manipulate metadata for the data objects
  - obtain URI through which the content of the data can be accessed
- So the ideas are the same as WebDAV.
- What would a service that implements both look like?

# Storage Differences

- ContainerNode <-> WebDAV collection
- DataNode
  - UnstructuredDataNode <-> WebDAV resource
  - StructuredDataNode -- No equivalent in WebDAV
- Link Node <-> WebDAV Redirect (experimental RFC 4437)
- VOSpace properties <-> WebDAV properties

# Capabilities

- VOSpace allows you to query the server to find out various capabilities
- For the most part, WebDAV capabilities that we care about are all required by the RFC's. There is only the ability to query about some advanced capabilities (locking).



# Views

- VOSpace allows services to convert nodes on input and output
- WebDAV: you get what you put in. No more, no less

# Protocols

- When returning results, they both allow the service to return multiple endpoints with different protocols and let the client decide
- But if you do not provide an HTTP URL first, ordinary WebDAV clients will get confused.

# Access Control

- Both support
  - Anonymous
  - httpauth
  - x509
  - cookie
  - OAuth

# Manipulating Nodes

- All of these operations only work within one VOSpace
- The VOSpace methods can be asynchronous, while the HTTP/WebDAV are synchronous.
- createNode <-> HTTP PUT
- moveNode <-> WebDAV MOVE
- copyNode <-> WebDAV COPY
- deleteNode <-> HTTP DELETE

# Accessing Metadata

- getNode <-> WebDAV PROPFIND
- setNode <-> WebDAV PROPPATCH
- findNode <-> RFC 5323 (WebDAV SEARCH)

# Third Party Transfers

- VOSpace enables transfers between data centers without going through the client.
  - pushFromVoSpace
    - WebDAV's COPY allows arbitrary destination URI's.
  - pullToVoSpace
    - WebDAV's COPY could be simply extended to allow arbitrary URI's in the source.
- VOSpace is asynchronous, while WebDAV is synchronous

# Identifiers

- VOSpace defines vos: identifiers which must be resolved
- WebDAV uses plain HTTP(S), so resolution is via the usual DNS mechanism.
- So a WebDAV server has to resolve any vos: identifiers it would be serving or just not generate a vos: URI at all.

# Summary

- Functionality that will not exist in WebDAV without significant work
  - StructuredNodes and Views
  - Asynchronous operations
  - vos: URI's
  - capabilities
- Not in WebDAV but in other standards
  - linkNode -> RFC 4437
  - findNode -> RFC 5323
- Third party pull transfers are a simple protocol extension to WebDAV